

Fast Marching in motion planning for rhombic like vehicles operating in ITER

Javier V. Gómez¹, Alberto Vale², Filipe Valente², João Ferreira², Santiago Garrido¹ and Luis Moreno¹

Abstract—Rhombic like vehicles are characterized for high maneuverability in cluttered environments. This type of vehicles will be used on remote handling operations of maintenance in the International Thermonuclear Experimental Reactor (ITER). Previous work was done in motion planning using Constrained Delaunay Triangulation for rhombic like vehicles operating in ITER. This paper shows that the integration of Fast Marching Square improves the motion planning methodology, decreasing also the computational effort, which can be applied not only in ITER but also in other complex and cluttered environments. Simulated results are presented comparing the initial and the improved motion planning.

I. INTRODUCTION

The world's rising demand for energy is a key issue in the near future, since current energy sources are either finite (fossil fuels), or their output is insufficient to meet demand. It is in this context that the International Thermonuclear Experimental Reactor (ITER) project was born, which will act as an experimental facility to prove the feasibility of fusion power as an alternative, safe and reliable source of energy.

The reactor will be installed inside the Tokamak Building (TB) of ITER, as illustrated in Figure 1. The remote handling (RH) systems are required during maintenance operations that will play an important role in the ITER project, [1]. One of such systems is the Cask and Plug Remote Handling System (CPRHS), a mobile vehicle responsible for RH operations of transportation of contaminated components and equipment from the TB to the Hot Cell Building (HCB). The largest CPRHS has dimensions 8.5m x 2.62m x 3.7m (length, width, height) and when fully loaded weights approximately 100 tons. The CPRHS is divided into three main components: the Cask, that contains the load; the Pallet, that supports the Cask; and the Cask Transfer System (CTS). The CTS acts as a mobile robot, by driving the entire vehicle, or by moving independently from the other components. The CTS has a rhombic kinematic configuration, as described in [2] and depicted in Figure 2. This configuration allows to control the velocity, V_i , and orientation, θ_i , of each wheel $i \in \{R, F\}$. Additionally the rhombic configuration provides the ability for both wheels to follow the same path, in this paper referred as line guidance, or for each wheel to follow a different path,

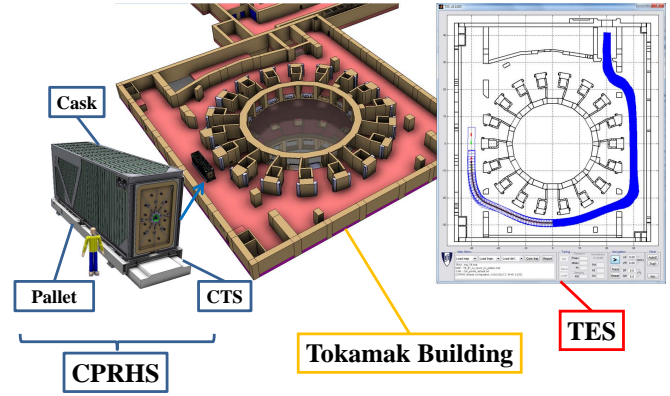


Fig. 1. Left - CAD model of the Cask and Plug Remote Handling System. Middle - Level B1 of the Tokamak Building with the Cask and Plug Remote Handling System in operation. Right - Snapshot of the software application to evaluate the trajectory optimization.

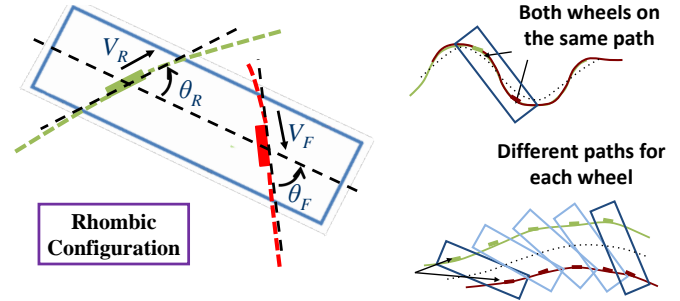


Fig. 2. Left - Model of the rhombic vehicle and the Cask Transfer System control variables. Right - Possible motion options for a rhombic like vehicle.

referred as free roaming providing flexibility when moving in the cluttered environments of ITER.

To perform the required RH operations, the vehicle must move along optimized trajectories and for that purpose a motion planning methodology was developed, [3]. This methodology is divided in three steps. In the first step the initial geometric path is obtained by using Constrained Delaunay Triangulation (CDT) [4]. In the second step an offline path optimization is performed inspired on the elastic bands theory, [5], which maximizes obstacle clearance and path smoothness, while minimizing path length. In the third step the trajectory is evaluated by determining the vehicle's speed profile along the path.

However, the initialization path obtained with CDT methodology presents some limitations in terms of path smoothness. These limitations can be overcome by using the Fast Marching Square (FM²) method [6], providing a good initialization and then, reducing the computation time needed to generate a feasible trajectory.

¹ J.V. Gómez, S. Garrido and L. Moreno are with RoboticsLab, Carlos III University of Madrid, Av. de la Universidad 30, 28911, Madrid, Spain {jvgomez,sgarrido,moreno} at ing.uc3m.es

² A. Vale, F. Valente and J. Ferreira are with Instituto de Plasmas e Fusão Nuclear, Instituto Superior Técnico, Universidade Técnica de Lisboa, Av. Rovisco Pais 1, 1049-001 Lisboa, Portugal {avale,filipe.valente,jftferreira} at ipfn.ist.utl.pt

The paper is organized as follows: Section II introduces the motion planning methodology and describes the respective three steps. Sections III and IV explain the FM² and how it is introduced in the previous motion planning methodology. The Section V presents some simulated results and, finally, Section VI summarizes the main conclusions.

II. MOTION PLANNING

The vehicle is required to move along a path that simultaneously maximizes the clearance to obstacles and reducing the distance between the start and goal poses (position and orientation). The proposed motion planning methodology is based on a three step approach, [3], as illustrated in Figure 3:

Geometric path evaluation: given the environment model and initial and goal poses, an initial geometric path is found. At this point the aim is to find a path connecting the initial and goal objectives that can act as an initial condition for the next path optimization step.

Path optimization: this module receives the preceding geometric solution as input and returns an optimized path.

Trajectory evaluation: in this final module, the speed profile is defined along the optimized path transforming it into a trajectory, which is the output of the proposed motion planning methodology.

A. Geometric Path Evaluation

From the 3D CAD models, a 2D representation is obtained by projecting at floor level all the relevant elements that might conflict with the CPRHS. The levels of TB and HCB are well structured scenarios that can be modeled as a set of planar walls, whose footprint is a line segment and thus the 2D map can be considered as a set of line segments. The 2D map is decomposed into a set of triangles, by using CDT, to account for all walls. Afterwards the algorithm determines the set of sequence of triangles that contain and link the start and goal positions, as illustrated in Figure 3. Each sequence of triangles is then converted into a sequence of points (mid point of the common edge of two consecutive triangles) yielding a path. The shortest and feasible path is chosen as the geometric path, acting as the initial condition for the path optimization step.

B. Path Optimization

The initial geometric path does not guarantee a collision free path for a rigid body, such as the CPRHS, as shown in

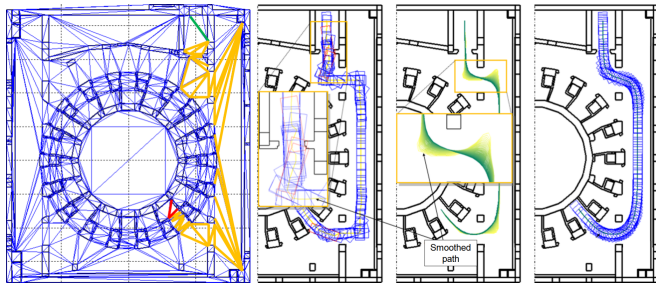


Fig. 3. From left to right: the initial map with the generated Constrained Delaunay Triangulation and the computed sequence of triangles between start and goal points, initial geometric path, path optimization and final optimized trajectory.

the second image of Figure 3 and thus may be unfeasible. Moreover this path is not smooth. To obtain an optimized path, two criteria are included in the algorithm [3]: clearance from obstacles, by increasing the distance from the vehicle to the walls and path smoothness, entailing getting shorter and smoother paths without slacks. To address the referred issues, the optimization step uses the elastic band concept [5], where the path is modeled as an elastic band, similar to a series of connected springs, subjected to two types of forces: internal and external forces. The first are the internal contraction forces, whose magnitude is proportional to the amplitude of displacement and determine that the path becomes retracted and shorter. The repulsive forces are responsible for keeping the path, and consequently the vehicle, away from obstacles. As detailed in [3], the kinematics of the vehicle are taken into account in the optimization step. Even though unfeasible trajectories may occur only given the maximum angle achieved by the orientation of the wheels and the respective velocities. The proposed method is able to identify these cases.

C. Trajectory Evaluation

The final trajectory is obtained by defining the speed of the vehicle at each point of the optimized path. In order to reduce the risk of collision in the case of major malfunction, the speed is reduced once the distance to the nearest obstacle decreases below a threshold value. Dynamic constraints, such as acceleration and speed limits, are also considered [3].

D. Geometric Path Evaluation Issues

A geometric representation of the environment has some advantages since it provides a very accurate representation of the scenario and it does not depend on grid cell discretization. However, in complex scenarios, the geometric representation results in some issues. In particular with the CDT, the representation requires a huge number of triangles, yielding to a computational effort. In addition, the CDT may result on sharp initial paths still far from the optimal one, as shown in Figure 11. As a consequence, the optimization takes longer. To overcome these issues, in this paper a new initialization method is proposed based on the Fast Marching Method (FMM) which is detailed in the next section.

III. INTEGRATION OF FAST MARCHING SQUARE

The FMM is a computational algorithm to solve the arrival time of expanding waves in every point of the space. Conceptually, it can be considered as a continuous version of the Dijkstra's algorithm [7]. It is based on the assumption that the information only flows outwards from the seeding area (wave source). The FM² method arises from the application of the FMM twice over the same map. The first time is used to create a map of velocities for the environment and the second time computes the time of arrival of the wave for every point when the wave is moving at the velocity computed in the previous step. The FM² method is very versatile when applied to motion planning problems. It has been successfully applied to many different problems such as autonomous exploration [8], outdoor motion planning [9] or even robot formation motion planning [10].

A. Fast Marching Method

The FMM was proposed by J. A. Sethian in 1996 to approximate the solution of the Eikonal equation, a non-linear partial differential equation encountered in problems of wave propagation, [11]. Let assume a 2D map, where $\mathbf{x} = (x, y)$ is a point on the map with the respective coordinates in relation to a Cartesian referential, the frontwave arrival time function for every point of the map, $T(\mathbf{x})$ and the velocity of the wave propagation $F(\mathbf{x})$ in each point \mathbf{x} . Let also assume that a wave starts propagating at $\mathbf{x}_0 = (x_0, y_0)$ at time $T(\mathbf{x}_0) = 0$ and with velocity $F(\mathbf{x}) \geq 0$. The Eikonal equation (1) defines the time of arrival the propagating frontwave $T(\mathbf{x})$ at each point \mathbf{x} of the map, in which the propagation speed depends on the point $F(\mathbf{x})$, according to:

$$|\nabla T(\mathbf{x})|F(\mathbf{x}) = 1 \quad (1)$$

Discretizing the gradient $\nabla T(\mathbf{x})$ according to [12] it is possible to solve the Eikonal equation at each point \mathbf{x} , which corresponds to the row i and column j of a grid map. Simplifying the notation as shown in (2), the equation (1) becomes (3):

$$\begin{aligned} T_1 &= \min(T(i-1, j), T(i+1, j)) \\ T_2 &= \min(T(i, j-1), T(i, j+1)) \end{aligned} \quad (2)$$

$$\left(\frac{T(i, j) - T_1}{\Delta x} \right)^2 + \left(\frac{T(i, j) - T_2}{\Delta y} \right)^2 = \frac{1}{F(i, j)^2} \quad (3)$$

The FMM consists on solving the equation (3) in which all the parameters are known, except $T(i, j)$. This process is iterative, starting at the source point of the wave (or waves) where $T(i_0, j_0) = 0$. The following iteration solve the value $T(i, j)$ for the neighbours of the points solved in the previous iteration. Using as an input a binary grid map, in which velocity $F(i, j) = 0$ (black) means obstacle and $F(i, j) = 1$ (white) means free space, the output of the algorithm is a map of distances as shown in Figure 4. These distances are concretely the time or arrival of the expanding wave at every point of the map.

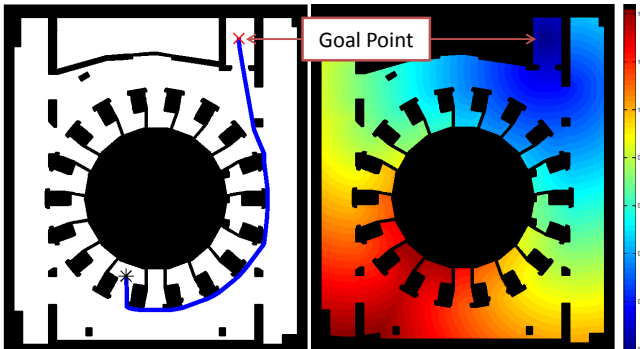


Fig. 4. Path (left) obtained with Fast Marching Method in the level B1 of Tokamak Building and the map of distances (right) obtained by means of Fast Marching Method.

FMM can be directly used as a path planner algorithm. By applying gradient descent from any point of the distance

map, a path will be obtained with the source of the wave as a goal point. This is valid only if one wave has been employed to generate the distance map. Otherwise, local minima will appear. The main advantage of this method is that the path obtained is optimal in distance. An example of a path provided by Fast Marching is shown in Figure 4.

B. Fast Marching Square

Although the paths provided by the FMM are optimal in distance terms, they do not accomplish the smoothness and safety constraints that most of robotic applications require. These paths run too close to obstacles and walls and have sharp curves. In view of these drawbacks, it turns out that the FMM algorithm is not enough in most of the situations. However, the FM^2 method [6] solves these two main disadvantages. The FM^2 method is based on automatically creating maps of velocities, $F(\mathbf{x})$, depending on the environment map in which the velocity of the expanding wave varies depending on the distance to the closest obstacle. The FMM is applied in order to obtain these maps of velocities. In this case, all the obstacles and walls are labeled as wave sources. Those waves expand with constant speed. The result is a map of distances in which those cells in the grid that are farther from the obstacles have a higher value, as depicted in Figure 5.

Once this map of distances is computed, it is normalized in order to have values between 0 and 1 and interpreted as relative wave expansion speeds (meaning full stopped or full speed of the wave expansion, respectively). The FMM is applied once again with the goal point as wave source. During the expansion, the wave will propagate with the velocities indicated in the map generated previously. The propagation ends once the initial point of the path is reached. The resulting a map of distances will be similar to the one obtained with the standard FMM, but with slight differences which make the paths very smooth when gradient descent is applied, as shown in Figure 6. In summary, FM^2 applies the FMM twice without any mathematical modification: the first step creates the map of velocities, $F(\mathbf{x})$, and the second step computes the time of arrival function, $T(\mathbf{x})$ in which

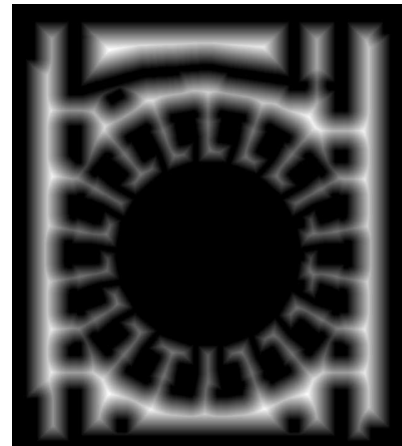


Fig. 5. Map of velocities obtained using all the black points of the grid map as a wave source in the Fast Marching Method.

gradient descent is applied to find the path. In FMM the map of velocities $F(\mathbf{x})$ is directly the input binary map.

In addition to the smoothness and safety, FM^2 yields:

- *No local minima*: as long as only one wave is employed to generate the map of distances, FMM ensures that there is a single global minimum at the source point of the wave (goal of the path).
- *Completeness*: the method finds a path if it exists and notifies in case of no feasible path.
- *Fast response*: The map of velocities is independent of the robot shape and pose. If the environment is static the map of velocities has to be calculated only once. Since the FMM method can be implemented with a complexity order of $O(n)$ [13], building the map of velocities is a fast process. In addition, using a binary grid map representation it is possible to jump some complex intermediate steps for creating the map of velocities.

IV. REPLACING GEOMETRIC PATH EVALUATION WITH FAST MARCHING SQUARE

The trajectory provided by FM^2 algorithm does not take into account the kinematics of the vehicle, neither it assures that it is feasible, since clashes might occur. However, the trajectory given by FM^2 algorithm is closer to a final solution, when compared to the initial geometric path obtained with CDT, as illustrated in the right image of Figure 7. Taking into account the workflow for trajectory optimization described in [3], the geometric path initialization using CDT can be replaced by the FM^2 , as illustrated in Figure 8.

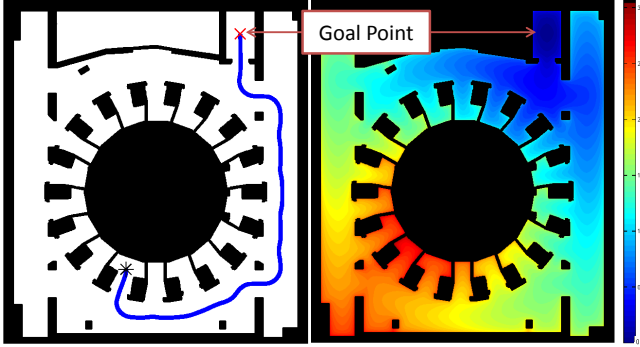


Fig. 6. Path (left) obtained applying gradient descent over the map of distances (right) obtained after applying Fast Marching Method to the map of velocities of Figure 5.

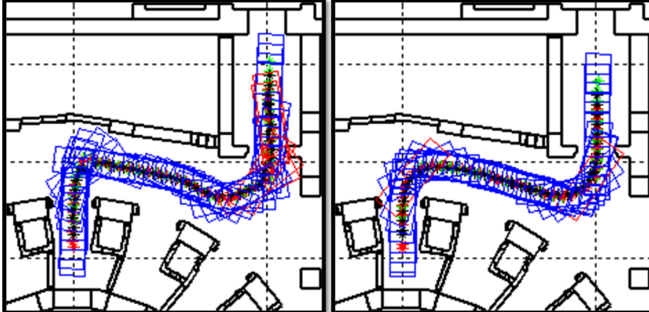


Fig. 7. Left - initial geometric path obtained with Constrained Delaunay Triangulation; Right - trajectory obtained with Fast Marching Square.

Since the FM^2 algorithm returns a single path to be followed by both wheels of the vehicle, comparisons will be made considering only line guidance. In addition, most of the nominal operations are accomplished using line guidance.

V. SIMULATED RESULTS

The simulated results presented in this section were evaluated using the software tool Trajectory Evaluator and Simulator (TES) shown in the right of Figure 1, which was developed under the field of grants F4E-2008-GRT-016 and F4E-GRT-276-01 to optimize trajectories for rhombic like vehicles in 2D maps of ITER buildings.

A. Simulations Settings

In order to assess the quality of the trajectories obtained using either CDT or FM^2 initializations a comparison criteria is defined. The trajectories are compared in terms of clearance, smoothness, computation time and number of iterations required in the optimization step.

In Figure 9 it is shown a schematic of the evolution of the path during the optimization step. The path connects the fixed initial and final points and has 3 additional points. The index n identifies the iteration of the optimization step.

At each iteration it is computed the variation of the path in relation to the previous iteration. The variation is computed evaluating the distance between each point of the path at iteration n and the line segment defined by the two nearest points of the path from iteration $n-1$, as illustrated in Figure 10. The 20 highest distances are selected and their median is computed. The stopping criteria is achieved when the median value is below a threshold (defined as 0.02 m) or if the number of iterations reaches 70.

B. Simulated Results and Analysis

The simulated results presented in this paper were obtained using the map of level B1 of TB, see Figure 1. The CDT provides a rough initial path mainly in the vicinity of the lift, highly subject to the shape of the triangles, as shown in Figure 11. The FM^2 method is independent to the geometric representation used in CDT to generate an initial path.

In Figure 12 it is presented a comparison between the initial and final trajectories using CDT and FM^2 , for port

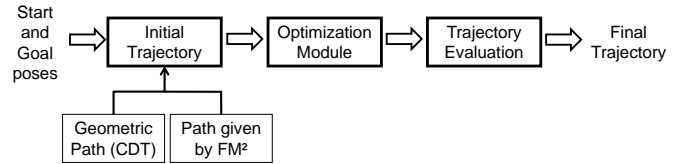


Fig. 8. Workflow for trajectory optimization.

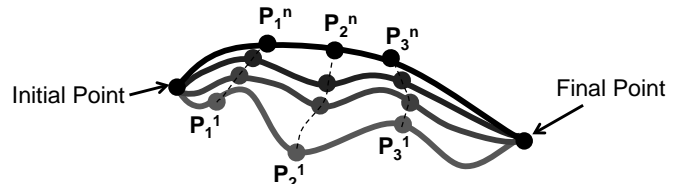


Fig. 9. Schema of the path evolution in each iteration during the trajectory optimization.

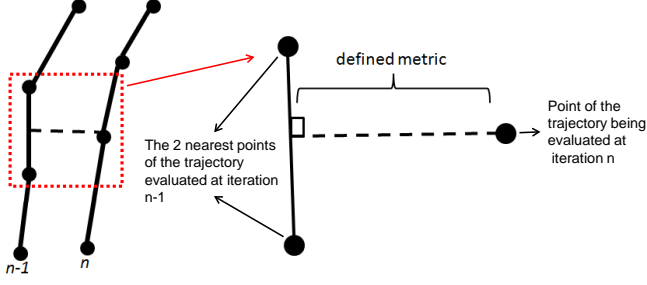


Fig. 10. Definition of the variation of the path between consecutive iterations: distance evaluated to a single point.

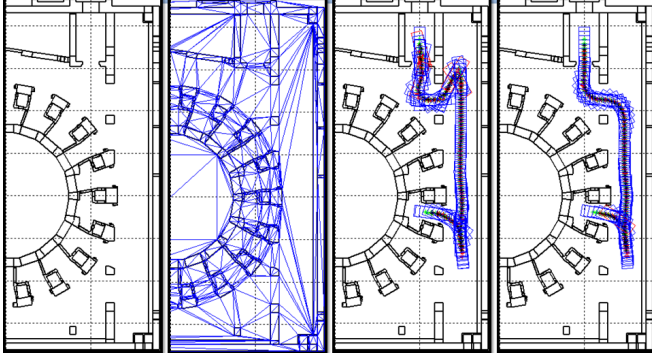


Fig. 11. From left to right: map of level B1 in Tokamak Building, generated Constrained Delaunay Triangulation of the map, geometric path obtained from Constrained Delaunay Triangulation and path obtained with Fast Marching Square.

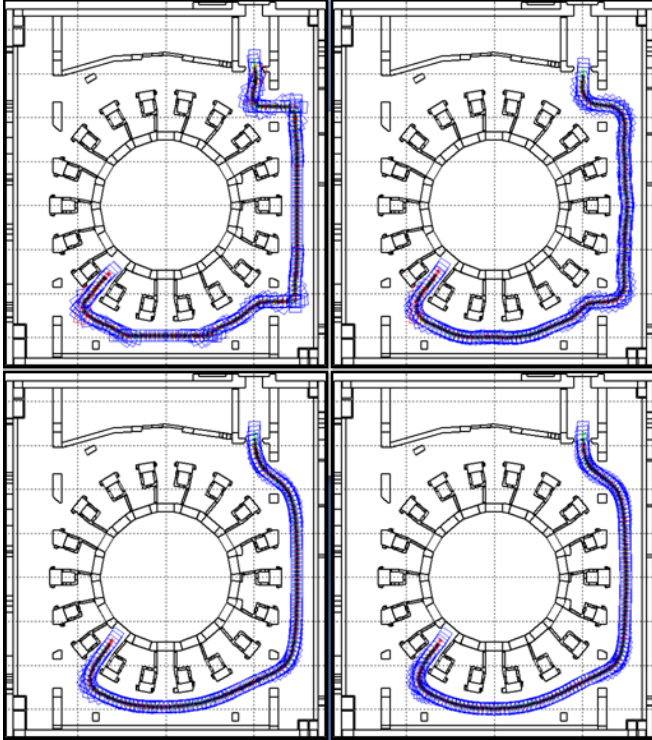


Fig. 12. Top: the initial trajectory for port 12 using as initialization the Constrained Delaunay Triangulation (left) and the Fast Marching Square (right); bottom: the final optimized trajectory for port 12 using as initialization the Constrained Delaunay Triangulation (left) and Fast Marching Square (right).

12. The map was slightly modified to reduce the effect of the triangulation in CDT in the vicinity of the lift, as illustrated in Figure 11. Initially both present clashes, but the initial trajectory using FM² is smoother. However, the final optimized trajectories using the CDT and FM² initialization are very similar as illustrated in Figure 12. The variation of the 20 highest distances and the respective median along the iterations are shown in Figure 13, where the methodology with the FM² initialization required less number of iterations to converge. The differences between the optimized trajectories can be identified when comparing the minimum distances to the closest obstacle, as depicted in Figure 14. The trajectory using FM² initialization in general has slightly higher obstacles clearance.

In Figure 15 it is presented the initial and final results for a trajectory to port cell 7. Once again the FM² initialization proves to be a smoother path and takes less iterations to converge when comparing with CDT initialization, see Figure 16. The minimum distance for the trajectory with FM² initialization also presents slightly higher values, as shown in Figure 17.

In Figure 18 it is presented a summary of the comparison of results between FM² and CDT initializations. In general the FM² initialization requires less computation time and number of iterations to converge, over the CDT initialization, which allows to decrease the computational effort that is needed for the optimization.

VI. CONCLUSIONS

This paper presented an improvement to the motion planning methodology used to compute the trajectory optimization in remote handling operations of ITER or in other cluttered environments. The geometric path initialization using

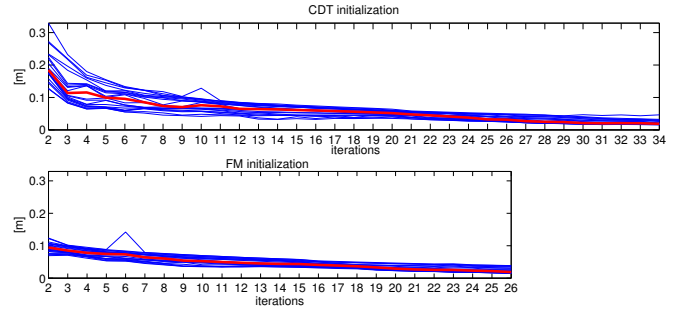


Fig. 13. Variation of the median (in red) along iterations for port 12 in level B1 of Tokamak Building.

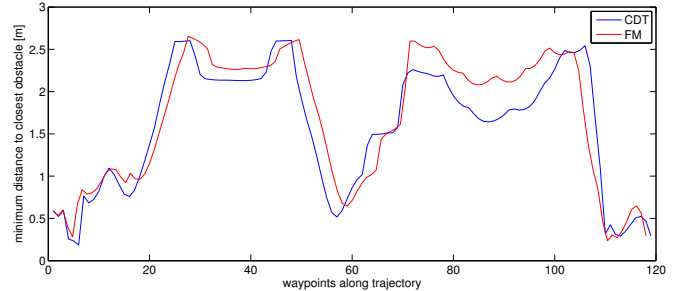


Fig. 14. Comparison between the minimum distances along the optimized trajectories using the Constrained Delaunay Triangulation and Fast Marching Square initializations for port 12 in level B1 of Tokamak Building.

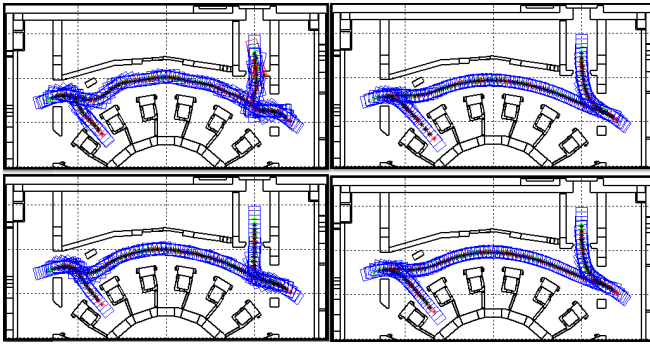


Fig. 15. Top: the initial trajectory for port 7 using as initialization the Constrained Delaunay Triangulation (left) and the Fast Marching Square (right); bottom: the final optimized trajectory for port 7 using as initialization the Constrained Delaunay Triangulation (left) and Fast Marching Square (right).

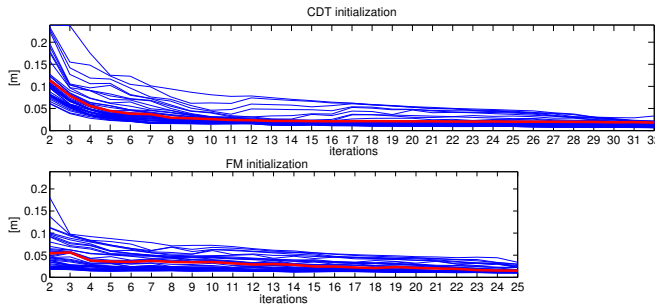


Fig. 16. Variation of the median along iterations for port 7 in level B1 of Tokamak Building.

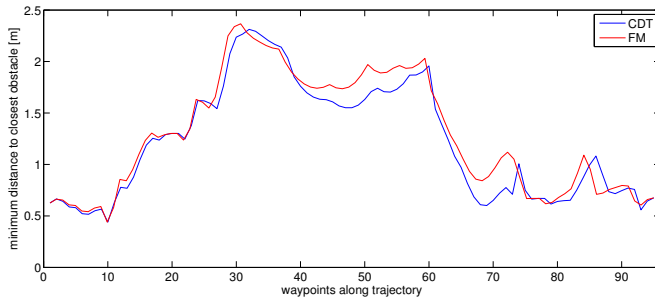


Fig. 17. Comparison between the minimum distances along the optimized trajectories using the Constrained Delaunay Triangulation and Fast Marching Square initializations for port 7 in level B1 of the Tokamak Building.

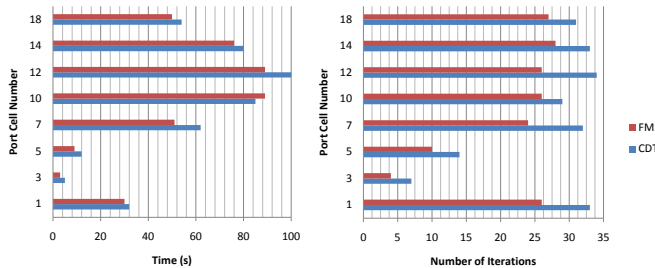


Fig. 18. Comparisons of computational time (left) and number of iterations (right) for trajectory optimization using Constrained Delaunay Triangulation and Fast Marching Square.

CDT was replaced by a powerful method: the Fast Marching Square. Given the simulated results, the optimization step is faster using the FM² when compared with the CDT. The FM² is also independent to the triangle representation, provides a smoother initial path and has no local minima. In terms of clearance the final trajectories with both methods are similar with small improvements when using the FM².

The proposed method is applicable to any dimension. However, the grid cell representation, in which FM2 is based, suffers an exponential increase of cells when the dimension increases. To overcome this issue, the total area of the scenario can be split into sub-areas and them, process each one individually. The proposed method becomes a divide and conquer approach using FM².

The paper was mainly focused to the ITER scenarios, where transfer operations are applicable. The proposed method can be used in any situation, in particular in very cluttered scenarios, which is crucial in ITER. The future work will focus on testing the algorithm in other scenarios, as warehouses.

A video summarizing the procedure and results of this paper can be found on <http://youtu.be/ShhX0hc0WR0>

ACKNOWLEDGMENT

The work was supported by FCT in the frame of the Contract of Associate Laboratories of Instituto de Plasmas e Fusão Nuclear/IST (Pest-OE/SADG/LA0010/2011) and by the project number DPI2010-17772 of the MICYT.

REFERENCES

- [1] I. Ribeiro, C. Damiani, A. Tesini, S. Kakudate, M. Siuko and C. Neri, "The Remote Handling Systems for ITER", *Fusion Engineering and Design*, vol. 86, pp. 471-477, 2011.
- [2] I. Ribeiro, P. Lima, R. Aparício and R. Ferreira, "Conceptual Study on Flexible Guidance and Navigation for ITER Remote Handling Transport Casks", *Proceedings of the 17th IEEE/NPSS Symposium on Fusion Engineering*, pp. 969-972, San Diego, USA (1997).
- [3] D. Fonte, F. Valente, A. Vale and I. Ribeiro, "A Motion Planning Methodology for Rhombic-like Vehicles for ITER Remote Handling Operations", *Proceedings of the 7th IFAC Symposium on Intelligent Autonomous Vehicles*, Lecce, Italy (2010).
- [4] L. P. Chew, "Constrained Delaunay Triangulations", *Proceedings of the Third Annual Symposium on Computational Geometry*, pp. 215-222, Waterloo, Ontario, Canada (1987).
- [5] S. Quinlan and O. Khatib, "Elastic Bands: Connecting Path Planning and Control", *Proceedings IEEE Conference Robotics and Automation*, vol. 2, pp. 802-807, Atlanta, USA (1993).
- [6] S. Garrido, L. Moreno, M. Abderrahim and D. Blanco, "FM²: A Real-time Sensor-Based Feedback Controller for Mobile Robots", *Intl. Journal of Robotics and Automation*, vol. 24, issue 1, pp. 3169-3192, 2009.
- [7] E. W. Dijkstra, "A Note on Two Problems in Connexion With Graphs", *Numerische Mathematik*, vol. 1, issue 1, pp. 269-271, 1959.
- [8] S. Garrido, L. Moreno and D. Blanco, "Exploration of a Cluttered Environment Using Voronoi Transform and Fast Marching", *Robotics and Autonomous Systems*, vol. 56, issue 12, pp. 1069-1081, 2008.
- [9] L. de Sanctis, S. Garrido, L. Moreno and D. Blanco, "Outdoor Motion Planning Using Fast Marching", *Proceedings of the Twelfth International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines*, pp. 1071-1080, Istanbul, Turkey (2009).
- [10] S. Garrido, L. Moreno and P. Lima, "Robot Formation Motion Planning Using Fast Marching", *Robotics and Autonomous Systems*, vol. 56, issue 9, pp. 675-683, 2011.
- [11] J. A. Sethian, "A Fast Marching Level Set Method for Monotonically Advancing Fronts", *Proceedings of the National Academy of Science*, vol. 93, issue 4, pp. 1591-1595, 1996.
- [12] S. Osher, J. A. Sethian, "Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations", *Journal of Computational Physics*, vol. 79, issue 1, pp. 12-49, 1988.
- [13] L. Yatziv, A. Bartesaghi and G. Sapiro, "A Fast O(n) Implementation of the Fast Marching Algorithm", *Journal of Computational Physics*, vol. 212, pp. 393-399, 2005.
- [14] J-C. Latombe, "Robot Motion Planning", *Kluwer Academic Publishers*, Boston, MA (1991).
- [15] S. M. LaValle, "Planning Algorithms", *Cambridge University Press*, (2006).